RPC BROKER

RELEASE NOTES

Version 1.1; XWB*1.1*40

September 1997

Revised February 2005

Department of Veterans Affairs
VistA Health Systems Design & Development (HSD&D)
Infrastructure and Security Services (ISS)

Revision History

Documentation Revisions

The following table displays the revision history for this document. Revisions to the documentation are based on patches and new versions released to the field.

Date	Revision	Description	Author
09/97	1.0	Initial RPC Broker Version 1.1 software release.	Thom Blom, San Francisco OIFO
02/19/02	2.0	Revised Version for Patch XWB*1.1*13.	Thom Blom, Oakland OIFO
02/22/05	3.0	Revised Version for Patch XWB*1.1*40 and previous undocumented patch updates.	Thom Blom, Oakland OIFO

Table i: Documentation revision history

Patch Revisions

For the current patch history related to this software, please refer to the Patch Module on FORUM.

Contents

RPC Broker V. 1.1 Release Notes

Overview

The Veterans Health Information Systems and Technology Architecture (VistA) Remote Procedure Call (RPC) Broker (also referred to as "Broker") Version 1.1, Patch XWB*1.1*40 is now available. This enhanced Broker software has the following functionality/features:

- Supports Single Sign-On/User context (SSO/UC)—As of Patch XWB*1.1*40, the TCCOWRPCBroker component enabled Single Sign-On/User Context (SSO/UC) in CCOW-enabled applications.
- Supports Non-Callback Connections—As of Patch XWB*1.1*35, the RPC Broker components are built with a UCX or non-callback Broker connection, so that it can be used from behind firewalls, routers, etc. This functionality is controlled via the new TRPCBroker component IsBackwardCompatibleConnection property.
- Supports Silent Logon capabilities—As of Patch XWB*1.1*13, the RPC Broker provides "Silent Login" capability. It provides functionality associated with the ability to make logins to a VistA M Server without the RPC Broker asking for Access and Verify code information.
- Documented Deferred RPCs and Capability to Run RPCs on a Remote Server.
- Multi-instances of the RPC Broker—As of Patch XWB*1.1*13, the RPC Broker code was modified to permit an application to open two separate Broker instances with the same Server/ListenerPort combination, resulting in two separate partitions on the server. Previously, an attempt to open a second Broker instance ended up using the same partition. For this capability to be useful for concurrent processing, an application would have to use threads to handle the separate Broker sessions.

Although we believe there should be no problems, the RPC Broker is not yet guaranteed to be thread safe.

- Updated components, properties, methods, and types.
- Separate Design-time and Run-time Packages—As of Patch XWB*1.1*14, the BDK contains separate run-time and design-time packages.
- Supports Delphi V. 6.0 and 7.0
- Operates in a 32-bit Microsoft Windows environment.

This version of the Broker also includes the Broker Development Kit (BDK). The BDK provides VistA application developers with the following features:

- The capability to create and implement client/server technology in the 32-bit Microsoft Windows environment using the Broker component (e.g., create Delphi-based client/server VistA applications with Graphical User Interfaces [GUI])
- Support for Commercial Off-the-Shelf (COTS) and Hybrid Open System Technology (HOST) client/server software using the Broker Dynamic Link Library (DLL)

Version 1.1 of the RPC Broker (fully patched) provides programmers with the capability to develop new VistA client/server software using the following RPC Broker Delphi components in the 32-bit environment (listed alphabetically):

- TCCOWRPCBroker
- TRPCBroker
- TSharedBroker
- TSharedRPCBroker
- TXWBRichEdit

These RPC Broker components wrap the functionality of the Broker resulting in a more modularized and orderly interface. Those components derived from the original TRPCBroker component, inherit the TRPCBroker properties and methods.

New Features and Enhancements of the RPC Broker

This version of the RPC Broker client/server interface provides the following *new* features and enhancements categorized by user type:

- **End Users** (e.g., clinicians)
- **System Managers** (e.g., IRM personnel)
- **Developers** (e.g., Programmers developing new VistA client/server programs in the 32-bit Microsoft Windows environment)

End Users—Features & Enhancements

Support for Single Signon/User Context (SSO/UC)

The Veterans Health Administration (VHA) information systems user community expressed a need for a single sign-on (SSO) service with interfaces to VistA, Health*e*Vet VistA, and non-VistA systems. This new architecture will allow users to authenticate and sign on to multiple applications that are CCOW-enabled and SSO/UC-aware using a single set of credentials, which reduces the need for multiple ID's and passwords in the Health*e*Vet clinician desktop environment. The RPC Broker software addressed this architectural need by providing a new TCCOWRPCBroker component in RPC Broker Patch XWB*1.1*40.

The TCCOWRPCBroker component allows VistA application developers to make their applications CCOW-enabled and Single Sign-On/User Context (SSO/UC)-aware with all of the client/server-related functionality in one integrated component. Using the TCCOWRPCBroker component, an application can share User Context stored in the CCOW Context Vault.

Thus, when a VistA CCOW-enabled application is recompiled with the TCCOWRPCBroker component and other required code modifications are made, that application would then become SSO/UC-aware and capable of single sign-on (SSO).

For more information on SSO/UC, please consult the *Single Sign-On/User Context (SSO/UC) Installation Guide* and *Single Sign-On/User Context (SSO/UC) Deployment Guide* on the VistA Documentation Library (VDL).

Support for Silent Logon

As of Patch XWB*1.1*13, the RPC Broker provides "Silent Login" capability. It provides functionality associated with the ability to make logins to a VistA M Server without the RPC Broker asking for Access and Verify code information.

Two types of Silent Login are provided with Version 1.1 of the BDK:

1. **Access/Verify Code-based—**One type of Silent Login uses Access and Verify codes provided by the application. This type of Silent Login may be necessary for an application that runs as a background task and repeatedly signs on for short periods. Another case would be for applications that are interactive with the user, but are running under conditions where they cannot provide a standard dialogue window, such as that used by the Broker to request Access and Verify codes. Examples might be applications running on handheld devices or within a browser window.
2. **Token-based—**The second type of Silent Login utilizes a token obtained by one application that is passed along with other information as a command line argument to a second application that it is starting. The token is obtained from the VistA server and remains valid for about twenty (20) seconds. When the newly started application sends this token during login the server identifies the same user and completes the login.

Due to the various conditions under which Silent Logins might be used, it was also necessary to provide options to the applications on error handling and processing. Applications that run as system services will crash if they attempt to show a dialogue box. Similarly, applications running within Web browsers are not

permitted to show a dialogue box or to accept windows messages. Properties have been provided to permit the application to handle errors in a number of ways.

As a part of the Silent Login functionality, the TVistaUser class providing basic user information was added. This class is used as a property by the TRPCBroker class and is filled with data following completion of the login process. This property and its associated data is available to all applications, whether they are using a Silent Login or not.

System Managers—Feature & Enhancements

On the Server

Management of Single Sign-On/User Context (SSO/UC)

Disabling SSO/UC

For sites whose policy is *not* to allow the kinds of SSO-based logins supported by SSO/UC, the User Context-based SSO can be disabled by doing either of the following:

- **Mark the User subject as "unshared" in the Sentillion Vergence Context Vault so that the User subject instance is kept separate for all application instances.** This is how the Sentillion Vergence Context Vaults were initially configured when VHA first procured them for Patient Context (i.e., User Context was specifically disabled).
- **Do *not* grant secure access in the Sentillion Vergence Context Vault to the application passcode used by the login components.** Without the application passcode, the login components *cannot* establish a secure binding to the User Context. This failure triggers a standard, non-SSO login process:
    1. The login component does not find a User Context.
    2. The login component prompts the user for their Access and Verify code credentials.
    3. The application logs in; and no User Context is set.

Kernel CCOW Login Token Expiration

The Kernel CCOW login token is valid from a minimum of 600 seconds to a maximum of 28,800 seconds (i.e., 10 minutes to 8 hours) from when the user first authenticated via Kernel on the VistA M Server. The default value is 5,400 seconds (i.e., 1.5 hours). This default value is a compromise between wanting to provide as rapid a Kernel CCOW login token expiration as possible for security reasons, versus the need for a SSO session to last long enough in order to be useful to the user.

To change the expiration time, IRM can change the value stored in the CCOW TOKEN TIMEOUT field (#30.1) in the KERNEL SYSTEM PARAMETERS file (#8989.3).

Management of Silent Logon

Control of the Silent Logon functionality is maintained and administered on the server for both VistA client/server applications (i.e., GUI) and the roll-and-scroll environment (i.e., terminal sessions).

Full Backward Compatibility with Broker V. 1.0

Broker-based applications compiled with Broker V. 1.0 (e.g., PCMM) will continue to work with this version of the Broker server routines.

On the Client

32-Bit Processing

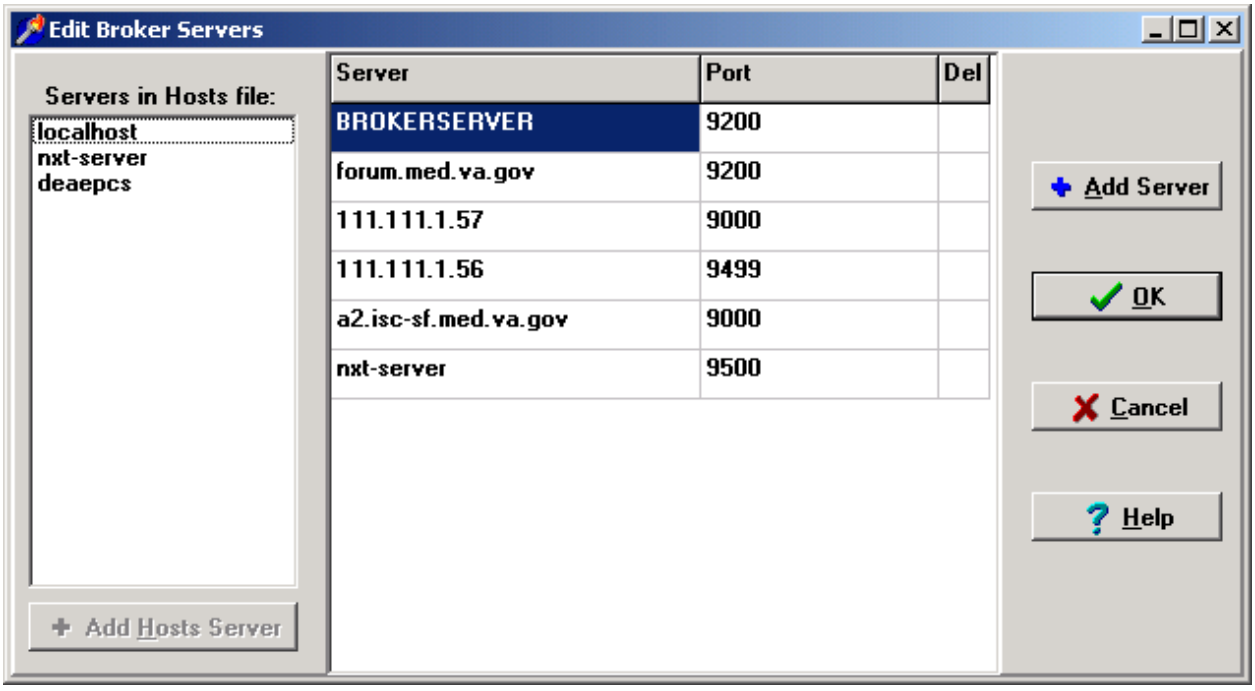
This version of the Broker operates in a 32-bit Microsoft Windows environment (i.e., client workstations running Microsoft Windows 2000 or XP operating systems). However, this version of the Broker will continue to support VistA applications previously developed for the 16-bit environment (e.g., PCMM).

Both the RPCBI.DLL and Client Manager (i.e., CLMAN.EXE) distributed with version 1.0 of the Broker are no longer required with this version of the Broker.

However, the RPCBI.DLL and Client Manager should *not* be removed. Previous 16-bit Broker-based applications (e.g., PCMM) still require the RPCBI.DLL and Client Manager.

Edit Broker Servers Program

If IRM wishes to add, modify, or delete servers and ports to be used by the Broker, run the Edit Broker Servers program (i.e., ServerList.EXE) to modify or add Listeners/Ports to the Windows Registry. ServerList.EXE can be copied to any workstation for this purpose.



The ServerList.EXE program only displays HOSTS file entries; it does not edit the HOSTS file.

Adding Entries:

You are given two methods of adding new server/port entries to the Registry:

1. Add a server listed in your HOSTS file
2. Add a new server that is automatically validated via the Domain Name Service (DNS) or the HOSTS file

Modifying Entries:

In order to modify or change a server or port, simply place the cursor in the appropriate Server or Port field and make the change.

Deleting Entries:

In order to delete a pre-existing entry, just click in the Del column. An asterisk appears in the Del column signifying a deletion. Another click toggles the deletion off.

Developers—Features & Enhancements

The 32-bit version of the Broker Development Kit (BDK) is now available and included with the release of version 1.1 of the RPC Broker.

For those developers who received the preview BDK released in October of 1995, many of the following features are either new or enhancements to the preview BDK.

This topic highlights some of the major changes made to the RPC Broker V. 1.1 since its original release (patch references are included where applicable):

- 32-Bit Processing/Delphi
- RPC Broker Components
  - Classes Added
  - Library Methods Added/Modified
  - Properties Added
  - Types Added/Modified
- Design-time and Run-time Packages
- Modified GetServerInfo Function
- Updated Dynamic Link Library (DLL) Interface
- Source Code Availability

32-Bit Processing/Delphi

This version of the Broker operates in a 32-bit Microsoft Windows environment only (i.e., client workstations running Microsoft Windows 2000 or XP operating systems). It provides programmers with the capability to develop new VistA client/server software using the Broker Delphi component (i.e., TRPCBroker) in the 32-bit environment. To develop VistA applications in a 32-bit environment you must have Delphi V. 2.0 or greater. This version of the RPC Broker component will *not* allow you to develop applications in Delphi V. 1.0. However, the Broker routines on the M server will continue to support VistA applications previously developed and compiled for the 16-bit environment (e.g., PCMM).

Please note that this version of the Broker supports Delphi V. 6.0 and 7.0.

RPC Broker Components

Version 1.1 of the RPC Broker (fully patched) provides programmers with the capability to develop new VistA client/server software using the following RPC Broker Delphi components in the 32-bit environment (listed alphabetically):

- TCCOWRPCBroker
- TRPCBroker
- TSharedBroker
- TSharedRPCBroker
- TXWBRichEdit

These RPC Broker components wrap the functionality of the Broker resulting in a more modularized and orderly interface. Those components derived from the original TRPCBroker component, inherit the TRPCBroker properties and methods.

For a complete description of the RPC Broker components, properties, and methods, please refer to the "RPC Broker Developer's Guide" (i.e., BROKER.HLP).

TCCOWRPCBroker

As of Patch XWB*1.1*40, the TCCOWRPCBroker component was added to Version 1.1 of the RPC Broker. The TCCOWRPCBroker Delphi component allows VistA application developers to make their applications CCOW-enabled and Single Sign-On/User Context (SSO/UC)-aware with all of the client/server-related functionality in one integrated component. Using the TCCOWRPCBroker component, an application can share User Context stored in the CCOW Context Vault.

Thus, when a VistA CCOW-enabled application is recompiled with the TCCOWRPCBroker component and other required code modifications are made, that application would then become SSO/UC-aware and capable of single sign-on (SSO).

TRPCBroker

The original TRPCBroker Delphi component provides Delphi developers with an easy, object-based access to the Broker. It is compatible with the Delphi object oriented (OO) environment. This component, when placed on a Delphi form, allows you to connect to the server and reference M data within Delphi's Integrated Development Environment (IDE). It makes a Delphi form and everything on it "data aware."

TSharedBroker

As of Patch XWB*1.1*26, the TSharedBroker component was added to Version 1.1 of the RPC Broker. The TSharedBroker Delphi component provides applications or plugins to applications easy access to an RPCBroker without the need for a separate M partition. Each component has its own security (i.e., option) as well. The default value of the AllowShared is True. If an application will have RPCs that require extensive time, it would be best to NOT share a Broker instance and the AllowShared should then be set to False.

TSharedRPCBroker

As of Patch XWB*1.1*26, the TSharedBroker component was added to Version 1.1 of the RPC Broker. The TSharedRPCBroker Delphi component provides applications or plugins to applications easy access to an RPCBroker without the need for a separate M partition. Each component has its own security (i.e., option) as well. The default value of the AllowShared is True. If an application will have RPCs that require extensive time, it would be best to NOT share a Broker instance and the AllowShared should then be set to False.

TXWBRichEdit

As of Patch XWB*1.1*13, the TXWBRichEdit component was added to Version 1.1 of the RPC Broker. The TXWBRichEdit Delphi component replaces the Introductory Text Memo component on the Login Form. TXWBRichEdit is a version of the TRichEdit component that uses Version 2 of Microsoft's RichEdit Control and adds the ability to detect and respond to a Uniform Resource Locator (URL) in the text. This component permits us to provide some requested functionality on the login form. As an XWB namespaced component we are required to put it on the Kernel tab of the component palette, however, it rightly belongs on the Win32 tab.

Classes Added

The following Classes were added to or modified in Version 1.1 of the RPC Broker:

- TVistaLogin

  As of Patch XWB*1.1*13.

- TVistaUser

  As of Patch XWB*1.1*13.

- TXWBWinsock

  As of Patch XWB*1.1* 40.

Library Methods Added/Modified

The following Library Methods were added to or modified in Version 1.1 of the RPC Broker.

Added

As of Patch XWB*1.1*13, the following library methods were new in the TVCEdit Unit:

- ChangeVerify

  ```
  function ChangeVerify(RPCBroker: TRPCBroker): Boolean;
  ```

- SilentChangeVerify

  ```
  function SilentChangeVerify(RPCBroker: TRPCBroker; OldVerify, NewVerify1,
  NewVerify2: String; var Reason: String): Boolean;
  ```

- StartProgSLogin

```
procedure StartProgSLogin(const ProgLine: String; ConnectedBroker: TRPCBroker);
```

As of Patch XWB*1.1*40, the following library methods were new with the TCCOWRPCBroker component:

- GetCCOWtoken

```
function GetCCOWtoken(Contextor: TContextorControl): string;
```

- IsUserCleared

```
function IsUserCleared: Boolean;
```

- IsUserContextPending

```
function IsUserContextPending(aContextItemCollection: IContextItemCollection):
Boolean;
```

- WasUserDefined

```
function WasUserDefined: Boolean;
```

Modified

As of Patch XWB*1.1*13, the following library methods were modified:

- CheckCmdLine

```
function CheckCmdLine(SLBroker: TRPCBroker): Boolean;
```

  Changed from procedure to function with a Boolean return value.

- GetServerInfo

  The GetServerInfo library function in the RpcConf1unit, which can be used to select the desired Server name and ListenerPort, was modified to add a "new" button. This button can be used to add a new Server/ListenerPort combination to those available for selection. It will also accept and store a valid IP address, if no name is known for the location. This will permit those who have access to other Server/ListenerPort combinations that may not be available in the list on the current workstation to access them. However, they will still need a valid Access and Verify code to log on to the added location.

- TParams

  procedure Clear was moved from Private to Public.

- TRPCB Unit

```
TOnLoginFailure = procedure (VistaLogin: TVistaLogin) of object;
```

  Changed from Object: TObject, since this is what should be expected by the procedure if it is called.

```
TOnRPCBFailure = procedure (RPCBroker: TRPCBroker) of object;
```

Changed from Object: TObject, since this is what should be expected by the procedure if it is called.

Properties Added

The following Properties were added to or modified in Version 1.1 of the RPC Broker:

- **TCCOWRPCBroker Properties:**
  - CCOWLogonIDName (Public)
  - CCOWLogonIDValue (Public)
  - CCOWLogonName (Public)
  - CCOWLogonNameValue (Public)
  - CCOWLogonVpid (Public)
  - CCOWLogonVpidValue (Public)
  - Contextor (Public)

  As of Patch XWB*1.1*40.
- **TRPCBroker Properties**
  - BrokerVersion (Public)
  - CurrentContext (Public)
  - IsBackwardCompatibleConnection (Published)
  - IsNewStyleConnection (Public)
  - KerneLogIn (Published)
  - LogIn (Public)
  - OldConnectionOnly (Published)
  - OnRPCBFailure (Public)
  - RPCBError (Public)
  - ShowErrorMsgs (Published)
  - User (Public)

  As of Patches XWB*1.1*13 and 35.
- **TSharedBroker and TSharedRPCBroker Properties:**
  - AllowShared (Public)
  - OnConnectionDropped (Public)
  - OnLogout(Published)

  As of Patch XWB*1.1*23.

- **TVistaLogin Properties:**
  - DomainName (Public)
  - IsProductionAccount (Public)

  As of Patch XWB*1.1*40.
- **TVistaUser Property:**
  - Vpid (Public)

  As of Patch XWB*1.1*40.

Types Added/Modified

As of Patch XWB*1.1*13 and XWB*1.1*40, the following Types were added to or modified in Version 1.1 of the RPC Broker:

- TLoginMode
- TShowErorMsgs
- TOnLoginFailure
- TOnRPCBFailure
- TParamType

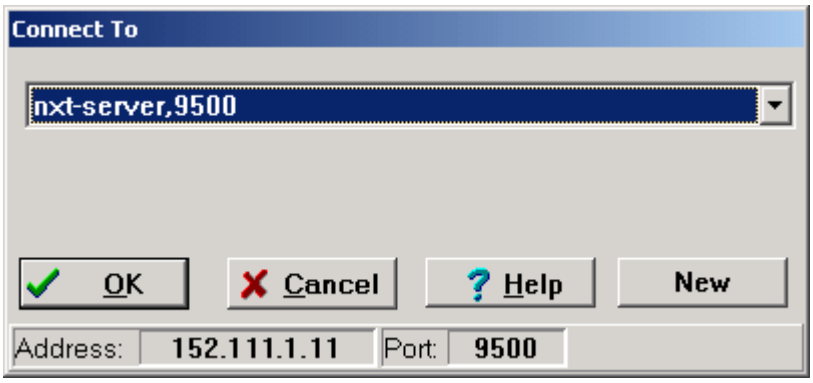
Design-time and Run-time Packages

As of Patch XWB*1.1*14, the BDK contains separate run-time and design-time packages.

Modified GetServerInfo Function

The GetServerInfo function obtains the end-user's target server and port. Use this function to set the TRPCBroker component's Server and ListenerPort properties before connecting to the server.

If there is more than one server/port to choose from, GetServerInfo displays an application window that allows users to select a service to connect to:



Updated Dynamic Link Library (DLL) Interface

This version of the Broker provides an updated set of Dynamic Link Library (DLL) functions that allow applications written in *any* Microsoft Windows-based development environment (e.g., Borland's Delphi, Borland C++, Microsoft Visual Basic, and other COTS products), to take advantage of all the features offered by the Broker component. This reflects VistA's continued movement toward open systems that support multiple GUI and client front-ends.

The Dynamic Link Library (DLL) functions act like a "shell" around the Delphi TRPCBroker component and provide developers with an easy function-based access to the Broker component. These functions allow GUI and client front-end applications written in Borland's Delphi and other COTS products to take advantage of all the features that the Broker offers. All of the communication to the server is handled by the TRPCBroker component accessed via the DLL interface.

The BAPI32.DLL contains all of the 32-bit Broker DLL functions. It provides an interface to the Broker component.

Source Code Availability

As of Patch XWB*1.1*14, the BDK contains the Broker source code. The source code is located in the ..\BDK32\Source directory.

Modified BDK source code should *not* be used to create VistA GUI applications.

Not all methods and properties found in the source code are documented at this time. Only those documented methods and properties are guaranteed to be made backwards compatible in future versions of the BDK.

Reference Materials

Updated RPC Broker Context-sensitive Online Help

This version of the Broker provides updated online context-sensitive help (i.e., BROKER.HLP) for the RPC Broker-related components and associated DLL exported procedures and functions. Select the component on a form, or highlight a particular procedure or function, and press the F1 key to get help on that item.

The online help also includes other related topics for IRM and the Broker developer (e.g., Tutorials, RPC information, Troubleshooting and Debugging tips, etc.).

Updated RPC Broker Documentation

This version of the Broker provides a full set of updated documentation. Readers who wish to learn more about the RPC Broker should consult the following:

- *RPC Broker Release Notes* (this manual)
- *RPC Broker Installation Guide*
- *RC Broker Systems Manual*
- *RPC Broker Technical Manual*
- *RPC Broker Getting Started with the Broker Development Kit (BDK)*
- *RPC Broker Developer's Guide* (i.e., BROKER.HLP, designed for programmers)
- RPC Broker Home Page at the following Web address:

  http://vista.med.va.gov/broker/index.asp

  This site contains additional information and documentation (e.g., Frequently Asked Questions [FAQs]) available in Hypertext Markup Language (HTML).

VistA documentation is made available online in Microsoft Word format and in Adobe Acrobat Portable Document Format (PDF). The PDF documents *must* be read using the Adobe Acrobat Reader (i.e., ACROREAD.EXE), which is freely distributed by Adobe Systems Incorporated at the following web address:

http://www.adobe.com/

For more information on the use of the Adobe Acrobat Reader, please refer to the *Adobe Acrobat Quick Guide* at the following web address:

http://vista.med.va.gov/iss/acrobat/index.asp

VistA documentation can be downloaded from the Health Systems Design and Development (HSD&D) VistA Documentation Library (VDL) Web site:

http://www.va.gov/vdl/

VistA documentation and software can also be downloaded from the Enterprise VistA Support (EVS) anonymous directories:

- Albany OIFO ftp.fo-albany.med.va.gov
- Hines OIFO ftp.fo-hines.med.va.gov
- Salt Lake City OIFO ftp.fo-slc.med.va.gov
- Preferred Method download.vista.med.va.gov

  This method transmits the files from the first available FTP server.